Collective Operations over multiple layers/groups

Krita has the ability to store multiple layers/groups in a clipboard and paste them. But it could not apply cut, copy, paste or delete a selection over multiple layers/groups. The project aims to create a user friendly way to perform these operations over multiple layers/groups with a single key-stroke. Here is a related bug and video for reference.

Current Scenario:

- Krita cannot perform deletion of the selected area on a group layer.
- Krita saves only paint data of a layer (or layers collectively) inside a selection to the clipboard while copy/cut.
- While paste, it creates a new single layer with paint data saved in the clipboard.

Currently Krita saves only paint data of a layer (or layers collectively) inside a selection to the clipboard while copy/cut. This project proposes to save a new layer (or layers) containing the respective paint data.

Future Work:

- Ability to apply transforms and filters over multiple groups/layers.
- Ability to apply paste and delete operation over multiple frames of an animation.
- Designing UI carefully to prevent clash between operations over multiple layers and multiple frames.

Project Goals

- Ability to delete a selection in all layers of a group when performed on a group layer.
- When a selection is cut/copied, store the layer (or layers individually) with paint data over the selected region in the clipboard.
- While pasting (Ctrl+V) paste all layers in the clipboard.
- Add a new paste factory to paste everything in a new single layer (just the way it does now) with an option in the menu.
- Ability to apply above operations over multiple groups/layers.
- Working on the project, keeping in mind about extending these operations over multiple frames in future.

Implementation

1. Deletion in multiple layers

Currently deletion is not applied when *currentNode()* is a group. We should make deletion capable to apply over group layers as well as over selected layers.

- Modify *deleteSelection()* so as to apply deletion over all the selected nodes. For example in *KisTool*:

```
void KisTool::deleteSelection()
{
    KisNodesList nodes = selectedNodes();
    Q_FOREACH (KisNodeSP node, nodes) {
        KisResourcesSnapshotSP resources =
                                        new KisResourcesSnapshot(image(), node,
this->canvas()->resourceManager());

        if (!blockUntilOperationsFinished()) {
            Return;
        }
        if (!KisToolUtils::clearImage(image(), resources->currentNode(),
resources->activeSelection())){
            KoToolBase::deleteSelection();
        }
    }
}
```

- Modify *clearImage()* so that it could be applied to *KisGroupLayer* as well

```
bool clearImage(KisImageSP image, KisNodeSP node, KisSelectionSP selection)
{
    if (node->inherits("KisGroupLayer") {
        KisNodeSP child = node->firstChild();
        while (child) {
            clearImage(image, child, selection);
            child = child->nextSibling();
        }
    } else if (node && node->hasEditablePaintDevice()) {
        // perform deletion on a single paint layer.
    }
}
```

2. Cut/Copy of multiple layers

Modify *KisCutCopyActionFactory* which uses ActionHelper that uses *KisClipboard::instance->setClip()*, so that it creates new copies of selected nodes and use *KisClipboard::instance()->setLayers()* to save nodes itself.

- Instead of saving image data to the clipboard, create a new node and set the *KisPaintDevice* of the new layer with this paint device.
- In the case of *KisGroupLayer* perform the above procedure recursively as shown in the 'deletion' section. The newly created group should be structurally similar to the original group layer.
- Gather all the created nodes and use *KisClipboard::instance()->setLayers()* to save nodes to the clipboard.

3. Pasting multiple layers/groups

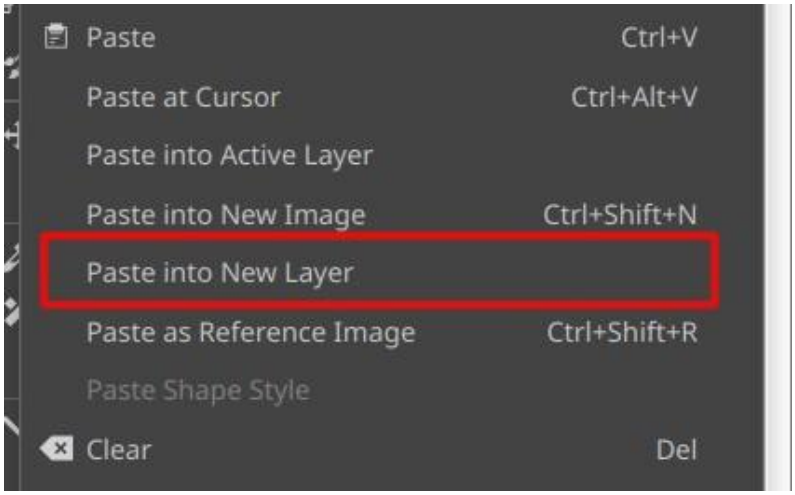
We should set *KisPasteActionFactory* to implement *KisNodeManager:: pasteLayersFromClipboard()* while performing default paste action (Ctrl+V).

4. Retaining existing features

- Paste into a single layer

  Currently while performing paste operation, the clipboard data is pasted into a new single layer. This project creates a new *KisPasteSingleLayerActionFactory* that retrieves all layers in the clipboard, creates a new merged layer, and adds it to the image.

  UI: This is performed by a newly added option in "Edit" options.



- Paste into active layer and Paste at Cursor

  This is also performed as above, but the difference is that the paint data is retrieved from the newly created ‘merged’ layer and added to the active layer.

  UI: This already has an option in “Edit” options.

- Copy Merged

  Currently, *KisCopyMergedActionFactory* uses *ActionHelper* to save image data. Instead we create a new single layer with the same image data and save it to the clipboard.

- Copy Sharp and Cut Sharp

  These features are set to be retained in the modified *KisCutCopyActionFactory* described in the “Cut/Copy of multiple layers” section.

Timeline

Community Bonding Period (May 17 - June 6)

This period is dedicated to:

- Communicate with the whole community and improve the implementation plans (both UI and codebase wise).
- Obtain a better understanding of the codebase of Krita relevant to the project (especially *KisClipboard* and *KisMimeData*).
- Learn more about Unit testing.
- Fixing bugs and improving docs that catch my sight.
- Write a blog post.

Deliverables Timeline

1. **Implementing deletion in multiple layers/groups (June 7 - June 21)**
   1.1. Modify existing *deleteSelection()* and other changes
   1.2. User testing and debugging
   1.3. Code Documentation
   1.4. Make a merge request and get feedback
   1.5. Feature Improvements and merge to main branch
   1.6. Write a feature blog post

2. **Copy/Cut saving layers to *KisClipboard* (June 22 - July 7)**
   2.1. Work on *KisCutCopyActionFactory* and other factories

2.2. Modify *ActionHelper*
2.3. Code Optimization and Documentation
2.4. Make a merge request and get feedback

3. **Testing and debugging period (July 8 - July 16)**
   3.1. Feature testing and improvements
   3.2. 1st phase evaluation

4. **Pasting and retaining existing features (July 17 - August 6)**
   4.1. Implement a new *KisPasteSingleLayerActionFactory*
   4.2. Modifying all affected feature factories
   4.3. Testing and Debugging (special focus)
   4.4. Documentation in code and manual
   4.5. Write a feature blog post

5. **Blogs and Docs (August 7 - August 13)**
   5.1. Final Wrap Up
   5.2. Complete any missing documentation wherever required
   5.3. Final Blog Post

If things go prior to the timeline, I am planning to start working on the mentioned "Future Work" regarding other operations on multiple groups/layers. I would start with applying transform operations over multiple groups/layers.

About me

I am Santhosh Anguluri, currently a pre-final year student of Bachelors degree in Computer Science and Engineering pursuing in Maulana Azad National Institute of Technology(MANIT), Bhopal. I am a competitive programmer participating in coding contests of codechef and codeforces. I am comfortable with c++, qt and python.

I have recently started contributing to open source software. KDE is the first open source organization I have started contributing to. I have been communicating with krita through IRC. Especially *halla*, *dmitryK*, *tiar*, *Wolthera_laptop* have been guiding me throughout my open source contributions. Following are some of my previous contributions to krita:

Fix rotation of ellipse and rectangle tools to vector layers
Rotation ability to rectangle and ellipse tools
Crop Tool: Ratio lock ignored while resizing beyond image border
Retaining dirty preset while renaming brush preset
416918 Fix "Default color model for new images"
431910 - Smoothing Options shortcuts for Multihand tool

Other overlapping schedules during GSoC: Exams of current semester will be finished by 2nd of May. Next semester, my college might start in the last week of July (news is still unconfirmed due to covid crisis). Though considering the worst case scenario, I can put myself in for 30 hours a week (at the least).

Communication: I will be in touch with the whole team in the IRC room daily. Throughout the coding period, I will write weekly blogs describing the week's work.

Contact Info:

Email: santhoshanguluri15@gmail.com

IRC Nick: *Santhosh_A* or *Santhosh_A[m]* (matrix logger for freenode)

Profiles:

Invent: https://invent.kde.org/santhoshanguluri

Github: https://github.com/SanthoshAnguluri

LinkedIn: https://www.linkedin.com/in/santhosh-anguluri-ba3707182/